# BENUTZERHANDBUCH

---

# Avisaro Compact Flash Box

"WISSEN WAS LÄUFT, SENDEN WAS LAUFEN SOLL"




Version 1.3

Änderungshistorie

| | | |
|---|---|---|
| 2004-10-01 | | Erstellung |
| 2004-10-25 | | At-Kommandos |
| 2004-11-04 | | Fehlermeldungen, Flash-Update, Quick-Start |
| 2004-11-18 | | Erweiterungen der Befehle und Geschwindigkeit Serielle Schnittstelle |
| 2004-12-13 | | Erweiterungen des Befehlssatzes. Insbesondere der Paketmodus ist hinzugekommen. Automatischer Lernmodus, ob bei der Eingabe <cr><lf> verwendet wird. Statusabfrage über AT+STATUS. Flusskontrolle (Hardware und Software) |

Kontakt:

Avisaro AG
Vahrenwalderstr. 7 (tch)
30165 Hannover

Telefon:
+49-511-9357411
Telefax:
+49-0721-151 254826

eMail:
support@avisaro.com

# INHALTSVERZEICHNIS

## EINLEITUNG

Mit der Avisaro Compact Flash Box haben Sie die Möglichkeit große Datenmengen auf handelsüblichen Speicherkarten zu speichern und ohne Umwege am PC zu lesen. Die Avisaro CF Box schließen Sie über eine übliche serielle Schnittstelle (RS232) an Ihre Anwendung an.

Um mit den Speichermedien zu arbeiten senden Sie Befehle über die serielle Schnittstelle. Zum „ausprobieren“ können Sie auch die CF Box an einen PC anschließen und dann an einem Terminal Programm die Befehl von Hand eingeben.

Der Ablauf um Daten zu schreiben oder zu lesen gliedert sich in drei Schritte:

1) Datei zum Schreiben oder Lesen vorbereiten (at+create, at+append, at+open)

2) Daten übermitteln (at+datastream, at+writepacket, at+readpacket)

3) Vorgang abschließen (at+close, Stop-Sequenz)

Darüber hinaus stehen noch Befehle zur Verwaltung des Datenträgers (Speichergröße, freier Speicherplatz) und zur Verwaltung des Moduls zur Verfügung (Uhrzeit).

## SICHERHEITSMAßNAHMEN

**Beachten Sie alle üblichen Sicherheitsmaßnahmen bei dem Betrieb von elektrischen Geräten.**

**Öffnen Sie niemals das Produkt. Führen Sie niemals Reparaturen oder Modifikationen selbst durch – überlassen Sie dies ausschließlich der Avisaro AG.**

## HAFTUNGSAUSSCHLUSS

**Die Avisaro AG haftet nicht für Schäden die durch den Einsatz der Avisaro Compact Flash Box entstehen. Insbesondere für den Ausfall des Produkts an sich und für Folgeschäden wie z.B. Betriebsunterbrechnungen oder Schäden an angeschlossenen Anlagen.**

**Ausgeschlossen ist auch die Haftung für die Vollständigkeit der Daten beim Schreiben und Lesen auf Speichermedien.**

# BETRIEB

## INBETRIEBNAHME

Schließen Sie die Avisaro Compact Flash Box über ein serielles Kabel an Ihre Anwendung an. Zur Stromversorgung benutzen Sie ein geeignetes Steckernetzteil, dass 5V bis 6V Spannung und mindestens 200mA Strom liefert. Die Avisaro CF Box ist sofort einsatzbereit und immer ‚eingeschaltet' so lange sie mit Strom versorgt wird.

Die serielle Schnittstelle wird im Auslieferungszustand mit 9600 Baud betrieben.

## SPEICHERKARTEN EINSTECKEN UND ENTNEHMEN

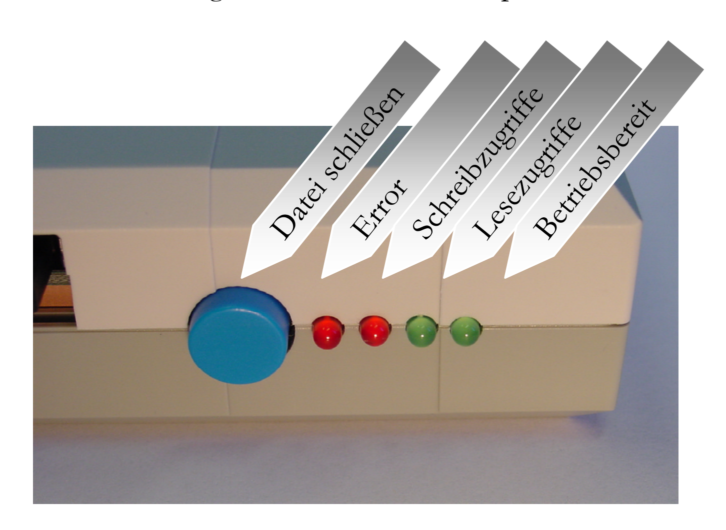
Speicherkarten können im eingeschaltetem Zustand eingesteckt werden. Die Avisaro Compact Flash Box erkennt automatisch eine neue eingesteckte Karte.

Die Karte kann im eingeschaltetem Zustand entnommen werden. **Dabei müssen Sie darauf achten, dass keine Schreiboperation stattfindet.** Sie erkennen aktive Schreiboperationen an der Leuchtdiode am Gehäuse.

Je nach Anwendung empfiehlt es sich mit dem „at+cardpresent" Befehl zu überprüfen, ob eine Karte eingesteckt ist bevor mit Schreib- und Lesebefehlen begonnen wird.

## LAUFENDER BETRIEB

Im Betrieb werden über die serielle Schnittstelle Befehle zum lesen und schreiben von Dateien gegeben. Leuchtdioden am Gehäuse zeigen Schreib- und Leseoperationen an.



### ANZEIGE: BETRIEBSBEREIT

Mit der Leuchtanzeige „Betriebsbereit“ wir angezeigt, dass das Avisaro Modul mit Strom versorgt und betriebsbereit ist. Wird das Avisaro Modul mit Strom versorgt, wird ein Selbsttest durchgeführt – deshalb leuchtet die Anzeige etwas verzögert auf.

### ANZEIGE: LESEZUGRIFFE

Mit der grünen Leuchte werden Lesezugriffe angezeigt. Wird eine Datei zum Lesen geöffnet, leuchtet die Anzeige dauerhaft. Sie erlischt kurz bei jedem Lesezugriff. Wird die Datei geschlossen, erlischt die Anzeige dauerhaft.

Auch wenn Lesezugriffe die Integrität des Dateisystems nicht gefährden, sollte die Speicherkarte nicht entnommen werden wenn diese Anzeige leuchtet.

### ANZEIGE: SCHREIBZUGRIFFE

Mit der roten Leuchtanzeige werden Schreibzugriffe auf die Speicherkarte angezeigt. Wird eine Datei zum Schreiben geöffnet, leuchtet die Anzeige dauerhaft. Sie erlischt kurz bei jedem Schreibzugriff. Beim Schreiben werden auch Daten gelesen – deshalb flackert die Anzeige für Lesezugriffe ebenfalls. Wird die Datei geschlossen, erlischt die Anzeige dauerhaft.

Schreibzugriffe beeinflussen die Integrität des Dateisystems. So lange die rote Anzeige leuchtet, sollte die Speicherkarte auf keinen Fall entnommen werden. Mit der blauen Taste kann die Datei manuell geschlossen werden.

### ANZEIGE: FEHLER / ERROR

Ist ein Fehler aufgetreten, leuchtet diese Anzeige dauerhaft. Fehler können z.B. falsche AT-Befehle oder ein voller Datenträger sein. Die Anzeige leuchtet so lange bis mit einem ‚at-readerror’ der Fehler quittiert worden ist.

### BEDIENELEMENT: DATEI SCHLIEßEN

Mit dem blauen Taster wird die laufende Operation abgeschlossen und eventuell noch offene Dateien geschlossen. Vor der Entnahme der Speicherkarte sollte weder die Anzeige für Schreib- oder Lesezugriffe leuchten.

## BESONDERHEITEN

Folgende Besonderheiten sollten Sie beim Betrieb der Avisaro Compact Flash Box beachten:

- Die Avisaro Compact Flash Box hat keine eigene Uhr. Sie können jedoch eine Uhrzeit und ein Datum mit AT-Befehlen einstellen. Diese wird dann z.B. beim Erzeugen von Dateien verwendet.
- Verwenden Sie für Ihre Anwendung nie den Dateinamen „aviup001.s19“. Ein solche Datei wird verwendet, um neue Software in das Modul zu laden („Flash-Update“).
- Verwenden Sie für Ihre Anwendung nicht den Dateinamen „outdata.log“ und „indata.log“. Diese Dateien werden verwendet, wenn der automatische Datalogging und Datareplay Modus verwendet wird.

- Das Avisaro Modul ‚lernt' ob eine Zeile mit <cr><lf> oder nur mit <lf> abgeschlossen wird. Dabei wird angenommen, dass die Eingabe des ersten Befehls dem verwendeten Schema entspricht. Die Information des Zeilenabschlusses ist für das Modul wichtig – insbesondere bei der Verwendung des Paketmodus.

## DIE SERIELLE SCHNITTSTELLE

Die serielle Schnittstelle lässt sich im Bereich von 1.200 bis 115.200 Baud konfigurieren.

**Hinweis auf möglichen Datenverlust:**

Die Schreibgeschwindigkeit zur Compact Flash hängt von verschiedenen Faktoren ab. Zum einen gibt es Qualitätsunterschiede zwischen Herstellern von Speichermedien. Wesentlicher ist jedoch die Menge der auf der Speicherkarte vorhandenen Dateien und wie stark diese „Fragmentiert" sind. Eine frisch formatierte Speicherkarte ist schneller als eine mit vielen Dateien.

Bei hohen Geschwindigkeit (z.B. 115.200 Baud) besteht daher die theoretische Gefahr, dass es zu Datenverlust kommt. Wenn die Daten auf der seriellen Schnittstelle schneller gesendet werden, als sie zur Karte geschrieben werden, gehen Bytes verloren. Das Avisaro CF Modul hat einen sehr großen Eingangsspeicher, um dies möglichst zu vermeiden. Sollte es dennoch vorkommen, so wird dieser Fehler erkannt und durch die rote LED am Modul angezeigt. Mit dem Befehl at+readerror kann überprüft werden, ob es sich um einen Speicherüberlauf (gleichbedeutend einem Datenverlust) handelt.

Maßnahmen, um einen Datenverlust zu vermeiden:

- Benutzen Sie leere und frisch formatierte Speicherkarten.
- Machen Sie kurze Pausen beim Senden. Diese entstehen häufig bereits automatisch, da die meisten Applikationen eine gewissen Rechenzeit haben, bevor Daten geschrieben werden.
- Benutzen Sie niedrige Baudraten.
- Benutzen Sie Software oder Hardware Flusskontrolle. Dies bietet 100% Schutz vor Bufferüberlauf.
- Benutzen Sie den Paketmodus. Dies bietet 100% Schutz vor Bufferüberlauf.

## KOMMANDOMODUS, DATENMODUS UND PACKETMODUS

Es werden drei Modi unterschieden:

1) Im Kommandomodus werden Befehle eingegeben. D.h. in diesel Modus sind die 'AT'-Befehl gültig und somit können alle Dateioperationen ausgeführt werden

2) Im Datenmodus werden die Daten übertragen. Der Datenmodus wird durch den Befehl at+datastream eingeschaltet und mit der Stop-Sequenz (drei Plus-Zeichen im zeitlichem Abstand) wieder ausgeschaltet.

3) Im Packetmodus werden ebenfalls Daten übertragen. Durch den Befehl at+writepacket wird für eine wohldefinierte Anzahl von Bytes in den Datenmodus geschaltet und danach automatisch wieder zurück in den Befehlsmodus.

Die Unterscheidung der drei Modi ist sinnvoll, da Daten nicht als Befehle interpretiert werden dürfen.

## KOMMANDOMODUS

- Im Kommandomodus können Befehle beginnend mit AT+ eingegeben werden.
- Ein Kommando endet entweder mit einem CR, einem LF oder zwei Zeichen, die entweder CR oder LF sein können. Das erste Kommando nach dem Einschalten oder Neustart hat eine Verzögerung von drei Sekunden, während der auf ein zweites Endekennzeichen gewartet wird (entweder CR oder LF). Dadurch merkt sich das CF-Modul, mit welchen Zeichen der Benutzer seine Kommandos abschliesst.
- Kommandos können gross- oder klein geschrieben werden.
- Kommandos mit einer Länge über 64 Zeichen (nach dem AT+) werden mit einem Fehler abgebrochen.
- Nach der Ausführung eines Kommandos wird ein OK<CR+LF> (Erfolgreich) oder ERROR<CR+LF> zurückgesendet.
- Nach der Ausführung eines Kommandos befindet sich das CF-Modul wieder im Kommandomodus.

## DATENMODUS

- In den  Datenmodus wird AT+DATASTREAM gewechselt.
- In den Datenmodus kann nur gewechselt werden, wenn eine Datei geöffnet ist (entweder zum Lesen oder Schreiben).
- Das CF-Modul sendet OK<CR+LF> zurück.
- Ist eine Datei zum Schreiben geöffnet, dann können jetzt Rohdaten zum CF-Modul gesendet werden.
- Ist eine Datei zum Lesen geöffnet, dann wird augenblicklich der Inhalt der Datei zum Benutzer-Terminal gesendet.
- Beim Schreiben in eine Datei bricht der Datenmodus ab, wenn ein Filesystem-Fehler aufgetreten ist. In diesem Fall wird ERROR<CR+LF> ausgegeben und die Datei geschlossen.
- Beim Lesen aus einer Datei bricht der Datenmodus ab, wenn die Datei komplett übertragen wurde oder wenn ein Filesystem-Fehler aufgetreten ist. Im Fehlerfall wird ERROR<CR+LF> ausgegeben und die Datei geschlossen.

- Beim Schreiben in eine Datei kann der Datenmodus nur durch die Eingabe von +++ (mit mindestens 0.5 Sekunden Abstand) abgebrochen werden (wenn kein Fehler aufgetreten ist).
- Das CF-Modul befindet sich nach dem Abbruch des Datenmodus' wieder im Kommandomodus.

PAKETMODUS

- In den Paketmodus wird mit AT+WRITEPACKET gewechselt.
- In den Paketmodus kann nur gewechselt werden, wenn eine Datei zum Schreiben geöffnet ist.
- Bei aktivem Software Flow Control (XON/XOFF) wird Paketmodus abgelehnt.
- AT+WRITEPACKET gibt OK<CR+LF> zurück wenn in den Paketmodus gewechselt wurde.
- Nach dem Wechsel in den Paketmodus muß das Paket gesendet werden. Das Paket ist wie folgt aufgebaut: 2 Bytes Längenangabe, 1...1536 Bytes Daten, 2 Bytes CRC-Wert.
- Alle 2-Byte Werte sind 'Little Endian' (Low Byte zuerst)
- Die Längenangabe bezieht sich auf die Daten d.h. Länge und CRC werden nicht mitgezählt.
- Der CRC-Wert ist optional. Ein CRC-Wert von 0 bedeutet dass kein CRC-Wert angegeben wurde. In diesem Fall wird das Paket ohne CRC-Check akzeptiert.
- Nach dem CRC-Check wird OK<CR+LF> oder ERROR<CR+LF> zurückgesendet. Genauere Angaben bekommt man im Fehlerfall mit AT+READERROR.
- Nachdem das Paket gesendet wurde befindet sich das CF-Modul wieder im Kommandomodus d.h. für ein weiteres Paket muss erneut AT+WRITEPACKET und ein Paket gesendet werden.

## DATENSTROM UND PAKETMODUS

Es stehen zwei Möglichkeiten zur Verfügung Daten zur Karte zu schreiben bzw. Daten von der Karte zu lesen. Beim „Datastream“ werden die Daten kontinuierlich zur Avisaro Box geschrieben, während sie beim „Paketmodus“ in Blöcken übertragen werden.

Vorteil des Paketmodus ist insbesondere die garantierte Datenübertragung. Es kann kein Datenverlust durch einen Bufferüberlauf entstehen, ebenso kann kein Datenverlust durch Fehler auf z.B. dem seriellen Kabel entstehen.

Welcher Modus einfacher zu Verwenden ist, hängt von der Anwendung ab. Wenn immer eine bekannte Datenmenge (z.B. ein Datagramm) übertragen wird, dann bietet sich der Packetmodus an,

da keine Stop-Sequenz (drei Plus-Zeichen mit Pause) generiert werden muss. Werden kontinuierlich Daten unterschiedlicher Menge geschrieben werden müssen, bietet sich der Datenstrom Modus an.

| | **at+writepacket** | **at+datastream** |
|---|---|---|
| Vorteil | • Datensicherung durch CRC<br>• Datenverlust ausgeschlossen | • Einfache Handhabung bei variablen Datenmengen |
| Nachteil | • Die Datenmenge muss vor dem Schreiben bekannt sein<br>• CRC Datensicherung erfordert Rechenaufwand | • Eine Stopsequenz muss erzeugt werden, um wieder in den Befehlsmodus zurück zu wechseln<br>• Keine besondere Datensicherung |

# TIPPS UND TRICKS (FAQ)

Bei Anregungen und Problemen wenden Sie sich bitte jederzeit an uns. Am besten erreichen Sie uns über E-Mail: support@avisaro.com . Wünschen Sie einen Rückruf, geben Sie bitte Ihre Telefonnummer und eine geeignete Rückrufzeit an.

Vielleicht helfen auch folgende Anregungen:

## ANTWORTZEITEN

Die Speichergrößen von Compact Flash Speicherkarten sind sehr groß geworden. Wenn Sie viele Daten gespeichert haben, werden die Antwortzeiten z.B. auf ein „at-delfile“ Befehl mitunter recht lang. Wenn Sie z.B. eine 60 Megabyte große Datei löschen, dauert dies einige Minuten.

## BESCHÄDIGTES DATEISYSTEM REPARIEREN

Wenn Sie ein Speichermedium aus der Avisaro CF Box entnehmen während ein Schreib- oder Lesevorgang in Betrieb ist, kann das Dateisystem beschädigt werden. Dies kann recht leicht wieder ‚repariert' werden:

Stecken Sie die Speicherkarte in Ihren Windows PC. Die Speicherkarte finden Sie unter einem Laufwerksbuchstaben im „Arbeitsplatz“. Mit rechte Maustaste , „Eigenschaften“ und dann die Lasche „Extras“ bekommen Sie die Möglichkeit zur „Fehlerüberprüfung“. Klicken Sie auf „Jetzt prüfen“ um den Datenträger zu überprüfen und zu reparieren.

.

## BEISPIELE

Hier finden Sie häufig verwendete Dateioperationen als Beispiele erläutert.

Zur Erklärung:

- <cr><lf> bedeutet „Carriage Return" und „Line Feed" – wird normalerweise übertragen, wenn Sie die ‚Enter' Taste an Ihrem Terminal drücken. Wenn Sie eine programmierbare Steuerung haben, müssen Sie die Zeichen generieren. In der Programmiersprache „C" sähe dies so aus: printf(„at+create logfile.txt \r\n"). Wenn Sie die Zeichen einzeln übertragen, wäre dies die Zahl 13 (oder hex 0x0d) für <cr> und die Zahl 10 (oder hex 0x0a) für <lf>.

- Die Avisaro Modul sind ‚tolerant'. Wenn sie anstelle von <cr><lf> nur eins von beiden senden, funktioniert dies auch.

### DATEI SCHREIBEN

Um eine neue Datei (z.B. „logfile.txt") zu erzeugen und mit Daten zu füllen, gehen Sie wie folgt vor:

Die Datei wird zunächst erzeugt mit:

Befehl: **`at+create logfile.txt`** `<cr><lf>`

Antwort: **`OK`** `<cr><lf>`

Anschließend wird die Datenübertragung vorbereitet mit:

Befehl: **`at+datastream`** `<cr><lf>`

Antwort: **`OK`** `<cr><lf>`

Sie können nun Daten senden (binär oder ASCII):

```
The quick brown fox jumps over lazy dog. <cr><lf>
Die Avisaro Compact Flash Box ist klasse. <cr><lf>
Testende.
```

Um die Dateiübertragung zu beenden, senden Sie die Stop-Sequenz (drei ‚+' Zeichen mit jeweils mehr als 0,5 Sekunden Pause dazwischen):

Befehl: **`+      +      +`**

Antwort: **`OK`** `<cr><lf>`

Die Datei wurde automatisch geschlossen und die Operation ist beendet.

## DATEI LESEN

Um eine Datei (z.B. „logfile.txt“) zu lesen, gehen Sie wie folgt vor:

Die Datei wird zunächst erzeugt mit:

Befehl: **at+open logfile.txt** <cr><lf>

Antwort: **OK** <cr><lf>

Anschließend wird die Datenübertragung vorbereitet mit:

Befehl: **at+datastream** <cr><lf>

Antwort: **OK** <cr><lf>

Der Dateiinhalt wird nun automatisch an Ihre Anwendung übertragen:

**The quick brown fox jumps over lazy dog. <cr><lf>**

**Die Avisaro Compact Flash Box ist klasse. <cr><lf>**

**Testende.**

Sie erkennen das Datenende daran, dass keine Daten innerhalb von 0,5 Sekunden mehr kommen. Um den Vorgang abzuschließen senden Sie die Stop-Sequenz:

Befehl: **+ + +**

Antwort: **OK** <cr><lf>

Die Datei wurde automatisch geschlossen und die Operation ist beendet.

## WARTUNG (SOFTWAREUPDATE)

Sie können die Software der Avisaro Compact Flash Box recht einfach auf dem neusten Stand halten. So rüsten Sie neue Befehle nach oder korrigieren eventuelle Fehler in der Software.

Die neuste Software finden Sie im Internet unter:

http://www.avisaro.com/html/speicher_box.html

So funktioniert der Software-Update:

- Laden Sie sich das gewünschte File auf Ihren PC. Entpacken Sie die Datei. Die Updates Files heissen alle "aviup001.s19". Am besten bewahren Sie also das ZIP-File auf, weil dieses einen bedeutsameren Dateinamen hat.
- Speichern Sie die Datei ("aviup001.s19") auf eine Compact Flash Speicherkarte. **Verwenden Sie ausschließlich frisch formatierte und leere Speicherkarten.** Die Datei sollte im obersten Verzeichnis liegen.
- Schalten Sie das Avisaro Modul aus und ziehen Sie stecken die Speicherkarte mit dem Updatefile hinein.
- Schalten Sie das Modul wieder ein.
- **Warten Sie mindestens 2 Minuten**. In dieser Zeit lädt das Avisaro Modul das Updatefile. Schalten Sie in dieser Zeit das Modul nicht aus. Es gibt keine besondere Rückmeldung, wenn der Updatevorgang fertig ist - sie können sich aber voll und ganz auf die 2 Minuten verlassen (länger schadet nicht).
- Schalten Sie das Modul aus, entnehmen die Speicherkarte mit dem Updatefile - das war's. (Vergessen Sie nicht die „aviup001.s19“ Datei auf der Speicherkarte zu löschen, sonst starten Sie den Updatevorgang immer wieder).

## ÜBERSICHT AT-KOMMANDOS

Allgemeine Hinweise:

- Alle AT-Befehle werden mit „Carriage Return“ (<cr>) und „Line Feed“ (<lf>) abgeschlossen. Der Befehl zum erzeugen einer Datei lautet also: at+create test.txt <cr><lf>. In der nachfolgenden Aufstellung wird auf das <cr><lf> verzichtet. Siehe auch die ASCII Tabelle am Ende dieser Dokumentation.
- Alle Antworten (z.B. ok, error, 32000) werden ebenfalls mit <cr><lf> abgeschlossen. Antworten wie z.B. die Anzahl der freien Speicherplätze wird im ASCII – Format zurück gegeben ('3','2','0','0','0').
- Groß und Kleinschreibung ist bei den Befehlen nicht relevant. Alle Antworten vom Modul werden immer in Großbuchstaben gegeben.
- Dateinamen haben das Format 8.3 (8 Zeichen Name und 3 Zeichen Kennung)
- Parameter werden mit „Leerzeichen“ von einander getrennt.

| DATEIEN ZUM SCHREIBEN UND LESEN VORBEREITEN | | |
|---|---|---|
| **Befehl** | **Antwort** | **Beschreibung** |
| **Datei zum Lesen öffnen:**<br>at+open *dateiname*<br>at+open *directory**dateiname* | OK, ERROR | Öffnet eine vorhandene Datei zum Lesen. Ist die Datei nicht vorhanden, wird ‚ERROR' zurück gegeben.<br>Um die Datenübertragung zu starten, wird der Befehl at+datastream verwendet.<br>Um den Vorgang abzuschließen, z.B. um eine andere Datei zu lesen, wird der Befehl at+close verwendet.<br>TIP: Der at+open Befehl kann auch verwendet werden, um zu testen ob eine Datei vorhanden ist: ‚OK': Datei ist da, ‚ERROR': Datei ist nicht da. Es muss nur nach einem ‚OK' der at+close Befehl geschickt werden, wenn die Datei nicht weiter verwendet werden soll.<br>**Voraussetzung:**<br>Es darf keine Datei geöffnet sein.<br>**Mögliche Rückgabewerte von at+readerror:**<br>- FILE IS OPEN |

| | | |
|---|---|---|
| | | - I AM HAPPY<br><br>- Filesystem Fehler (Nummer 0x40 bis 0x4d) |
| **Neue Datei erzeugen:**<br><br>at+create *dateiname*<br><br>at+create *directory**dateiname* | OK, ERROR | Erzeugt eine Datei auf dem Datenträger mit dem Namen „Dateiname“. Wird nur ein Dateiname angegeben, wird die Datei im Wurzelverzeichnis angelegt. Mit „\directory\dateiname“ wird eine Datei im angegebenen Verzeichnis angelegt. Dies muss vorhanden sein. Ist eine Datei mit dem selben Namen bereits vorhanden, so wird diese gelöscht.<br><br>Nach dem Erzeugen ist die Datei geöffnet. Soll sie nicht weiter verwendet werden, muss sie mit „at+close“ geschlossen werden.<br><br>**Voraussetzung:**<br><br>Es darf keine Datei geöffnet sein.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- FILE IS OPEN<br><br>- I AM HAPPY<br><br>- Filesystem Fehler (Nummer 0x40 bis 0x4d) |
| **Daten an Datei anhängen:**<br><br>at+append *dateiname*<br><br>at+append *directory**dateiname* | OK, ERROR | Es wird die vorhandene Datei “Dateiname” zum schreiben geöffnet. Alle Daten werden an das Ende der Datei angehängt.<br>Um Daten zu senden, kann der Befehl at+datastram verwendet werden.<br><br>**Voraussetzung:**<br><br>Es darf keine Datei geöffnet sein.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- FILE IS OPEN<br><br>- I AM HAPPY<br><br>- Filesystem Fehler (Nummer 0x40 bis 0x4d) |

| DATEN ÜBERTRAGEN | | |
|---|---|---|
| **Befehl** | **Antwort** | **Beschreibung** |
| **Datenstrom senden:**<br><br>at+datastream *daten* | OK, ERROR | Die Avisaro CF Box wird in den Daten-Modus versetzt. Im Datenmodus wird ein kontinuierlicher Datenstrom verarbeitet. Es muss zuvor eine Datei zum Schreiben geöffnet worden sein.<br><br>Alle gesendeten Daten werden in die geöffnete Datei geschrieben. Um den Vorgang zu beenden wird die „Stop-Sequenz“ (’+’,’+’,’+’) verwendet. Die Stop-Sequenz wird mit OK oder ERROR quittiert. Dabei wird die Datei geschlossen. Ein at+close ist nicht notwendig. Wenn ‚zwischendurch’ ein Fehler auftritt (z.B. Datenträger voll) wird ein „ERROR“ gesendet.<br><br>Daten können sowohl ASCII, wie auch binäres Format haben.<br><br>**Voraussetzung:**<br><br>Es muss eine Datei zum Schreiben oder Lesen geöffnet sein.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- FILE NOT OPEN<br><br>- I AM HAPPY |
| **Datenstrom lesen:**<br><br>at+datastream | OK, ERROR<br><br>Datenstrom | Die Avisaro CF Box wird in den Daten-Modus versetzt. Im Datenmodus wird ein kontinuierlicher Datenstrom verarbeitet. Es muss zuvor eine Datei zum Lesen geöffnet worden sein.<br><br>Die Daten aus der Datei werden ausgegeben. Um den Datenstrom anzuhalten kann „Software Flusskontrolle (XON, XOFF)“ oder „Hardware Flusskontrolle (RTS, CTS)“ verwendet werden. Um den Vorgang abzubrechen, kann die „Stop-Sequenz“ geschrieben werden. Der Vorgang ist beendet, wenn alle Daten aus der Datei übermittelt worden sind. Dieses Ende kann erkannt werden, wenn – bei abgeschalteter Flusskontrolle – keine Daten innerhalb von 0.5 Sekunden mehr kommen.<br><br>Die Datei wird automatisch geschlossen. Es ist also kein at+close Befehl notwendig, nachdem die Datei vollständig übertragen worden ist. |

| | | |
|---|---|---|
| | | Der at+datastream Befehl wird mit einem ‚OK' oder ‚ERROR' <cr><lf> bestätigt. Die Nutz-Daten aus der Datei beginnen demnach nach dem <cr><lf>.<br><br>Daten können sowohl ASCII, wie auch binäres Format haben. |
| **Daten als Paket schreiben:**<br><br>at+writepacket <cr><lf><br>*packet (length, data, crc)* | OK, ERROR | Anstelle eines kontinuierlichen Datenstroms werden Daten packetweise zum Schreiben übermittelt. Eine Datei muss vorher zum Schreiben geöffnet worden sein.<br>Das Moduls antwortet auf at+writepacket mit OK oder ERROR. Dann kann das Paket übertragen werden. Das Format ist:<br><br>• 2 Bytes: Datenlänge (nur Daten, ohne CRC). Das niederwertige Byte wird zuerst übertragen.<br><br>• x Bytes: Daten (Anzahl gemäß Länge, Maximal: 1536 Bytes.)<br><br>• 2 Bytes: CRC Prüfsumme. Die CRC Überprüfung kann ausgeschaltet werden, indem zwei Bytes mit einer 0 übertragen werden.<br><br>Das Avisaro Modul antwortet mit OK oder ERROR. Bei ‚ERROR' (z.B. falsche Prüfsumme) wurde das Paket nicht auf das Speichermedium geschrieben. Die Übertragung kann mit der Stop Sequenz abgebrochen werden. Die maximale Paketlänge ist durch das Modul vorgegeben (meist 512 Bytes).<br><br>**Voraussetzung:**<br><br>Funktioniert nicht, wenn XON/XOFF aktiv ist. Eine Datei muss vorher geöffnet worden sein.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- IMPROPER FLOW CONTROL METHOD<br><br>- FILE NOT OPEN<br><br>- I AM HAPPY |

| | | |
|---|---|---|
| **Daten paketweise lesen:**<br><br>at+readpacket<br><br>demnächst verfügbar | packet | Anstelle eines kontinuierlichen Datenstroms werden Daten paketweise gelesen. Eine Datei muss vorher zum Lesen geöffnet worden sein.<br>Das Modul antwortet mit Daten im Paketformat:<br>2 Bytes: Datenlänge (ohne CRC)<br>x Bytes: Daten (Anzahl gemäß Länge)<br>2 Bytes: CRC Prüfsumme<br>Mit einem wiederholten at+readpackt Befehl werden die nächsten Daten aus der geöffneten Datei angefordert.<br>Die maximale Paketgröße lässt sich konfigurieren. |
| **Daten wiederholen (lesen):**<br><br>at+repeatpacket<br><br>demnächst verfügbar | packet | Das letzte Datenpaket wird wiederholt. Dies wird verwendet, um z.B. bei einer fehlerhaften CRC Prüfsumme die Daten erneut anzufordern.<br>Das Modul antwortet mit Daten im obigen Paketformat: |

| **OPERATIONEN ABSCHLIEßEN UND BEENDEN** | | |
|---|---|---|
| **Befehl** | **Antwort** | **Beschreibung** |
| **Operation abschließen:**<br><br>at+close | OK, ERROR | Schließt eine geöffnete Datei. Dieser Vorgang ist wichtig, um die Integrität des Dateisystems zu gewährleisten. Der Befehl bezieht sich immer auf die aktuelle Datei, es werden also keine Parameter benötigt.<br><br>**Voraussetzung:**<br><br>Es muss eine Datei zum schreiben oder lesen geöffnet sein.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- FILE NOT OPEN<br><br>- I AM HAPPY<br><br>- Filesystem Fehler (Nummer 0x40 bis 0x4d) |
| **Stop-Sequenz:**<br><br>'+' .. Pause > 0.5 sec ..<br>'+' .. Pause > 0.5 sec ..<br>'+' | Je nach aktivem Befehl | Mit dieser Sequenz werden laufende Vorgänge beendet. Insbesondere wird so der Datenmodus beendet.<br>Die Sequenz setzt sich zusammen aus: drei ‚+' Zeichen mit jeweils einer Pause von mindestens 0.5 sec (höchstens 5 sec). Wird ein anderes Zeichen zwischendurch gesendet, wird die |

| | | |
|---|---|---|
| | | Sequenz abgebrochen.<br>Nach der Stop-Sequenz wird die Datei geschlossen. Sollen weitere Daten geschrieben/gelesen werden muss die Datei wieder geöffnet (at+append) werden.<br><br>Beim Schreiben (at+datastream) werden die + Zeichen nicht mit in die Datei geschrieben. |
| **XON**<br><br>= Hex-Wert 0x11 | keine | Setzt den Datenstrom fort. Aktiv beim Auslesen einer Datei und nachdem der Datenstrom mit XOFF pausiert wurde. |
| **XOFF**<br><br>= Hex-Wert 0x13 | keine | Hält den Datenstrom an. Aktiv beim Auslesen einer Datei.<br><br>Bei besetztem XOFF ist die Stop-Sequenz noch aktiv. |

| **DATENTRÄGER UND DATEIINFORMATIONEN** | | |
|---|---|---|
| **Befehl** | **Antwort** | **Beschreibung** |
| **Freier Speicherplatz:**<br><br>at+freespace | Anzahl der freien Bytes<br><br>OK, ERROR | Ermittelt den freien Speicherplatz auf dem Datenträger. Als Antwort wird die Zahl der freien Bytes zurück gegeben.<br><br>**Voraussetzung:**<br><br>Es darf keine Datei geöffnet sein.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- FILE IS OPEN<br><br>- I AM HAPPY<br><br>- Filesystem Fehler (Nummer 0x40 bis 0x4d) |
| **Größe des Datenträgers:**<br><br>at+diskspace | Anzahl der Bytes auf dem Datenträger<br>OK, ERROR | Ermittelt die Größe des Datenträgers.<br>Ist keine Speicherkarte eingelegt, wird ‚error' zurück gegeben |
| **Dateigröße:**<br><br>at+filesize | Dateigröße in Bytes<br><br>OK, ERROR | Ermittelt die Dateigröße. Die Datei muss vorher mit dem at+open Befehl geöffnet worden sein.<br><br>**Voraussetzung:**<br><br>Es muss eine Datei geöffnet sein.<br><br>**Mögliche Rückgabewerte von at+readerror:** |

| | | |
|---|---|---|
| | | - FILE NOT OPEN<br>- I AM HAPPY |
| **Dateidatum:**<br>at+filedate | Datum und Zeit der Datei im Format: jjjj/mm/tt hh:mm:ss | Ermittelt das Datum und die Zeit der Datei. Die Datei muss vorher mit dem at+open Befehl geöffnet worden sein. Die Angaben folgen im Abstand mit einer Leerzeichen („Space“).<br>**Voraussetzung:**<br>Es muss eine Datei geöffnet sein.<br>**Mögliche Rückgabewerte von at+readerror:**<br>- FILE NOT OPEN<br>- I AM HAPPY |

| **VERZEICHNISSE UND DATEIEN VERWALTEN** | | |
|---|---|---|
| **Befehl** | **Antwort** | **Beschreibung** |
| **Verzeichnis erzeugen:**<br>at+createdir *directory* | OK, ERROR | Es wird ein Verzeichnis mit dem Namen “Directory” angelegt. Ist das Verzeichnis bereits vorhanden, so wird eine Fehlermeldung ausgegeben.<br>**Voraussetzung:**<br>Es darf keine Datei geöffnet sein.<br>**Mögliche Rückgabewerte von at+readerror:**<br>- FILE IS OPEN<br>- I AM HAPPY<br>- Filesystem Fehler (Nummer 0x40 bis 0x4d) |
| **Inhaltsliste darstellen:**<br>at+dir \\<br>at+dir *directory* | Dateiname, Größe, Datum <cr><lf><br>Dateiname, Größe, Datum <cr><lf><br>ok,error<cr><lf> | Zeigt den Inhalt des Wurzelverzeichnisses oder eines Unterverzeichnisses an. Es wird eine Liste mit dem Format Dateiname Größe Datum zurückgegeben. Die Liste wird mit einem ‚OK’ oder ‚ERROR’ <cr><lf> abgeschlossen.<br>**Voraussetzung:**<br>Es darf keine Datei geöffnet sein.<br>**Mögliche Rückgabewerte von at+readerror:**<br>- FILE IS OPEN |

| | | |
|---|---|---|
| | | - I AM HAPPY<br><br>- Filesystem Fehler (Nummer 0x40 bis 0x4d) |
| **Datei / Verzeichnis löschen:**<br><br>at+delete *dateiname*<br><br>at+delete *directory* | OK, ERROR | Die Datei „dateiname“ oder das Verzeichnis „directory“ wird gelöscht. Ein zu löschendes Verzeichnis muss leer sein.<br><br>**Voraussetzung:**<br><br>Es darf keine Datei geöffnet sein.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- FILE IS OPEN<br><br>- I AM HAPPY<br><br>- Filesystem Fehler (Nummer 0x40 bis 0x4d) |
| **Datei verschieben:**<br><br>at+move d*ateiname dateiname*<br><br>demnächst verfügbar | OK, ERROR | |

| **BESONDERE BEFEHLE** | | |
|---|---|---|
| **Befehl** | **Antwort** | **Beschreibung** |
| **Karte vorhanden ?**<br><br>at+cardpresent | OK, ERROR | Es wird überprüft, ob eine Speicherkarte eingesteckt ist (OK) oder nicht (ERROR).<br><br>**Voraussetzung:**<br><br>Funktioniert unabhängig davon, ob eine Datei geöffnet ist oder nicht.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- MEMORY CARD NOT PRESENT<br><br>- I AM HAPPY |
| **Status abfragen**<br><br>at+status | Liste von Statusinformtionen<br><br>OK | Listet diverse interne Statusinformationen:<br>- Copyrightmeldung, Rev.Nr. etc.<br>- Eingabeerkennung zwei/ein Zeichen<br>- Art der aktiven Flow Control Methode<br>- Auslastung des Eingangspuffers<br>- Ob Flow-Control wirksam war<br>- Ob Bytes verschluckt wurden<br><br>**Voraussetzung:**<br><br>Funktioniert immer. |

| | | |
|---|---|---|
| | | **Mögliche Rückgabewerte von at+readerror:**<br><br>- I AM HAPPY |
| **Verfügbare Befehle / HELP**<br><br>at+commands | Liste der Befehle mit der Anzahl der erwarteten Parameter in Klammern.<br><br>OK | Listet alle verfügbaren Befehle und die Anzahl der erwarteten Parameter auf. Die Ausgabe erfolgt ohne das führende „at+“.<br><br>**Voraussetzung:**<br><br>Funktioniert immer.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- I AM HAPPY |
| **Mögliche Fehlermeldungen**<br><br>at+errorlist | Liste der Fehlermeldungen<br><br>OK | Listet alle verfügbaren Rückgabemeldungen des Befehls at+readerror.<br><br>**Voraussetzung:**<br><br>Funktioniert immer.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- I AM HAPPY |
| **Gesamt Speicherplatz**<br><br>at+diskspace | Speichergröße<br><br>OK, ERROR | Ermittelt die Größe des Datenträgers.<br>Ist keine Speicherkarte eingelegt, wird ‚error’ zurück gegeben. |
| **Fehlermeldungen lesen:**<br><br>at+readerror | Fehlernummer Fehlertext | Liest Detailinformationen zu einer Fehlermeldung aus. Es wird eine Fehlernummer und eine Fehlermeldung im Klartext ausgegeben. Abgeschlossen wird die Antwort mit <cr><lf>.<br><br>Es wird kein OK nach dem Fehlercode ausgegeben.<br><br>**Voraussetzung:**<br><br>Funktioniert unabhängig davon, ob eine Datei geöffnet ist oder nicht. |
| **Datum setzen:**<br><br>at+setdate *datum* | OK, ERROR | Setzt das Datum im Avisaro Modul. Dieser Befehl ist immer nach dem Einschalten notwendig, wenn das Avisaro Modul über keine batteriegepufferte Uhr verfügt. So lange das Modul eingeschaltet ist, wird Zeit und Datum weitergezählt. Das Datum wird verwendet, wenn Dateien erzeugt werden.<br>Das Datum wird im Format: jjjj mm tt |

| | | |
|---|---|---|
| | | übertragen (Leerzeichen dazwischen)<br><br>**Voraussetzung:**<br><br>Eingaben müssen numerisch sein d.h. dürfen keine Buchstaben haben. Tag und Monat werden auf gültigen Bereich geprüft.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- INVALID PARAMETER<br><br>- PARAMETER OUT OF RANGE<br><br>- I AM HAPPY |
| **Zeit setzten:**<br><br>at+settime *time* | OK, ERROR | Setzt die Zeit im Avisaro Modul. Dieser Befehl ist immer nach dem Einschalten notwendig, wenn das Avisaro Modul über keine batteriegepufferte Uhr verfügt. Die Zeit wird verwendet, wenn Dateien erzeugt werden.<br>Die Zeit wird im Format: hh mm ss übertragen. (Leerzeichen dazwischen).<br><br>**Voraussetzung:**<br><br>Eingaben müssen numerisch sein d.h. dürfen keine Buchstaben haben. Stunde, Minute und Sekunde werden auf gültigen Bereich geprüft.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>- INVALID PARAMETER<br><br>- PARAMETER OUT OF RANGE<br><br>- I AM HAPPY |
| **Modul neu starten:**<br><br>at+restart | | Das Modul wird neu gestartet. So werden z.B. die neu eingestellten Parameter für die serielle Schnittstelle übernommen.<br><br>**Voraussetzung:**<br><br>Funktioniert nur, wenn keine Datei geöffnet ist. Reset des Controllers über Watchdog-Timer. Ausgabe von OK nach dem Neustart.<br><br>**Mögliche Rückgabewerte von at+readerror:**<br><br>keine |
| **Daten eines Sectors:**<br><br>at+dump *sector* | | Listet die Daten eines Sectors aus. Diese Funktion dient zur Diagnostik im Fehlerfall. Detaillierte Kenntnisse von Dateisystemen sind |

| | | |
|---|---|---|
| | | erforderlich.<br>**Voraussetzung:**<br>Funktioniert nur, wenn keine Datei geöffnet ist. Gibt einen Sektor der FlashDisk aus (512 Bytes).<br>**Mögliche Rückgabewerte von at+readerror:**<br>- FILE IS OPEN<br>- I AM HAPPY |

| KONFIGURATION | | |
|---|---|---|
| **Befehl** | **Antwort** | **Beschreibung** |
| **Packetlänge:**<br>at+paketlength<br>demnächst verfügbar | OK, ERROR | Setzt die maximale Paketlänge beim Lesen einer Datei. |
| **<u>RS232 Paramter:</u>**<br>at+linepar *speed parity stop fluss*<br><br>**(Beachten Sie bei hohen Baudraten die Hinweise in Kapitel „Die serielle Schnittstelle“ auf Seite 7).** | OK, ERROR | Setzt die Parameter für die serielle Schnittstelle. Um diesen Befehl zu senden, muss mit den gegenwärtig eingestellten Parametern gearbeitet werden. Die Einstellung wird dauerhaft bis zum nächsten at+linepara gespeichert. Nach dem Ein-/Ausschalten bzw. nach „at+restart“ werden die neuen Parameter aktiv.<br>*Speed:* Baudrate: 1200, 2400,9600, 19200,38400,57600,115200<br>*Parity:* Parity: n (none), e (even), o (odd)<br>*Stop:* Anzahl Stop Bits: 1, 2<br>*Fluss:* Flusskontrolle: none, sw (software), hw (hardware)<br>Defaulteinstellungen sind: 9600 n 1 sw<br><u>Z. Z. wird nur der „Speed“ Parameter und die Flusskontroller ausgewertet.</u><br>**Voraussetzung:**<br>Es werden nur erlaubte  Parameter akzeptiert. Funktioniert unabhängig davon, ob eine Datei geöffnet ist oder nicht.<br>**Mögliche Rückgabewerte von at+readerror:**<br>- INVALID PARAMETER<br>- I AM HAPPY |

## FEHLERMELDUNGEN

| Nr. | Fehlertext | Beschreibung |
|---|---|---|
| 0x00 | I AM HAPPY | Es sind keine Fehler aufgetreten |
| 0x01 | INPUT BUFFER OVERFLOW | Es wurde ein überlanger Befehl eingegeben (mehr als 64 Zeichen). Verwenden Sie korrekten Syntax um dies zu vermeiden (z.B. schließen Sie Kommandos mit <cr><lf>. |
| 0x02 | PARAMETER COUNT MISMATCH | Es wurde ein Parameter vergessen. Z.B. bei at+open wurde der Dateiname vergessen. Wenn zu viele Parameter übergeben worden sind, so wird auch diese Fehlermeldung ausgegeben (z.B. wurde ein „Space“ zu viel eingegeben). |
| 0x03 | INVALID PARAMETER | Der übergebene Parameter stimmt nicht bzw hat ein falsches Format. Z.B. bei at+settime wurde ein fehlerhaftes Format verwendet. |
| 0x04 | PARAMETER OUT OF RANGE | Der übergebene Parameter ist zu groß oder zu klein. |
| 0x05 | COMMAND NOT IMPLEMENTED | Der verwendete Befehl steht noch nicht zur Verfügung. Schauen Sie auf der Web-Seite nach Software-Updates. |
| 0x06 | UNKNOWN COMMAND | Der verwendete Befehl ist unbekannt. Überprüfen Sie die Schreibweise. |
| 0x07 | FILE IS OPEN | Es ist eine Datei geöffnet und der Befehl kann nicht ausgeführt werden, so lange diese Datei offen ist. Z.B. ein at+open file1 gefolgt von einem at+open file2. Schließen Sie die Datei mit at+close und führen Sie den Befehl erneut aus. |
| 0x08 | NO OPEN FILE | Es wurde ein Befehl übertragen, der eine geöffnete Datei benötigt. Z.B. at+datastream benötigt einen at+open file Befehl vorab. |
| 0x09 | MEMORY CARD NOT PRESENT | Es ist keine Speicherkarte gesteckt, oder die gesteckte Speicherkarte wurde nicht als Speicherkarte erkannt. |
| 0x0A | LOW RS232 INPUT BUFFER | Der Eingangsspeicher zur Pufferung von Daten war gefährlich niedrig. Es ist zwar noch keine Datenverlust vorgekommen, aber es ist |

| | | |
|---|---|---|
| | | damit zu rechnen, dass dies passieren wird. |
| 0x0B | RS232 INPUT BUFFER OVERFLOW | Der Eingangsspeicher ist übergelaufen. Die Daten konnten nicht schnell genug zur Speicherkarte geschrieben werden. Es ist zum Datenverlust gekommen. |
| 0x0C | WRONG PACKET LENGTH | Paketmodus: Die Längenangabe ist kleiner als 1 oder größer als 1536 Bytes |
| 0x0D | CRC ERROR | Paketmodus: Der CRC-Wert ist falsch oder ungleich 0 (kein CRC-Wert). |
| 0x0E | IMPROPER FLOW CONTROL METHOD | Paketmodus: Software Flow Control (XON/XOFF) ist aktiv. Der Paketmodus akzeptiert nur Hardware Flow Control (RTS/CTS) oder kein Flow Control. |

| **Nr.** | **Fehlertext** | **Beschreibung** |
|---|---|---|
| 0x40 | ACCESS DENIED | Der Zugriff ist wurde verweigert. Z.B. ein at+create file , wärend file schreibgeschützt ist. |
| 0x41 | NAME CONFLICTS WITH SUBDIRECTORY | Es wurde versucht eine Datei zu erzeugen, deren Name als Verzeichnissname vorhanden ist. Wählen Sie einen anderen Namen. |
| 0x42 | INSUFFICIENT DIRECTORY SPACE | Es ist kein Speicherplatz mehr vorhanden. |
| 0x43 | DISK IMPROPERLY FORMATTED | Die Formatierung des Speichermediums ist fehlerhaft. Formatieren Sie das Medium auf einem PC neu, oder korrigieren Sie mit „chkdsk“ den Fehler. |
| 0x44 | ERROR IN READING OR WRITING DISK | Es konnte nicht geschrieben oder gelesen werden. Vielfältige Fehlerursachen. |
| 0x45 | CORRUPT FILE ALLOCATION TABLE | Das Dateisystem ist beschädigt. Fehlerursache ist meist, wenn die Karte herausgezogen wurde, während Daten geschrieben worden sind. |
| 0x46 | DIRECTORY DOES NOT EXIST | Das bezeichnete Verzeichnis ist nicht vorhanden. |
| 0x47 | INVALID DRIVE OR FILE NAME | Es können mehrere Partitionen auf dem Speichermedium sein. Dies wird zur Zeit noch nicht unterstützt.<br><br>Dieser Fehler wird auch ausgegeben, wenn ein ungültiger Pfad angegeben wurde. Wenn z.B. |

| | | |
|---|---|---|
| | | Verzeichnisse mit „/“ anstelle von „\“ getrennt eingegeben wurden. |
| 0x48 | INVALID MODE | Interner Softwarefehler. Es wurde ein falscher Modus verwendet. Bitte berichten Sie diesen Fehler an Avisaro (support@avisaro.com). |
| 0x49 | FILE NOT FOUND | Es wurde versucht auf eine Datei zuzugreifen, die nicht existiert. |
| 0x4a | FILE NAME OR DIRECTORY ALREADY EXISTS | Es wurde versucht ein Verzeichnis zu erzeugen, dass bereits existiert. Bitte wählen Sie einen anderen Namen. |
| 0x4b | SEEK OUT OF RANGE | Interner Softwarefehler. Der Dateizeiger hat einen falschen Wert. Bitte berichten Sie diesen Fehler an Avisaro (support@avisaro.com). |
| 0x4c | THE FILE CAN NOT BE RESTORED | Interner Softwarefehler. Der Dateizeiger hat einen falschen Wert. Bitte berichten Sie diesen Fehler an Avisaro (support@avisaro.com). |
| 0x4d | UNDEFINED FILE SYSTEM ERROR | Allgemeiner Fehler. Bitte wenden Sie sich an Avisaro zur Usachenforschung. |

## ASCII ZEICHEN TABELLE

Unter „www.asciitable.com“ finden Sie folgende und weitere nützliche ASCII Tabellen:

| Dec | Hx | Oct | Char | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr | Dec | Hx | Oct | Html | Chr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 000 | NUL (null) | 32 | 20 | 040 |   | Space | 64 | 40 | 100 | @ | @ | 96 | 60 | 140 | ` | ` |
| 1 | 1 | 001 | SOH (start of heading) | 33 | 21 | 041 | ! | ! | 65 | 41 | 101 | A | A | 97 | 61 | 141 | a | a |
| 2 | 2 | 002 | STX (start of text) | 34 | 22 | 042 | " | " | 66 | 42 | 102 | B | B | 98 | 62 | 142 | b | b |
| 3 | 3 | 003 | ETX (end of text) | 35 | 23 | 043 | # | # | 67 | 43 | 103 | C | C | 99 | 63 | 143 | c | c |
| 4 | 4 | 004 | EOT (end of transmission) | 36 | 24 | 044 | $ | $ | 68 | 44 | 104 | D | D | 100 | 64 | 144 | d | d |
| 5 | 5 | 005 | ENQ (enquiry) | 37 | 25 | 045 | % | % | 69 | 45 | 105 | E | E | 101 | 65 | 145 | e | e |
| 6 | 6 | 006 | ACK (acknowledge) | 38 | 26 | 046 | & | & | 70 | 46 | 106 | F | F | 102 | 66 | 146 | f | f |
| 7 | 7 | 007 | BEL (bell) | 39 | 27 | 047 | ' | ' | 71 | 47 | 107 | G | G | 103 | 67 | 147 | g | g |
| 8 | 8 | 010 | BS (backspace) | 40 | 28 | 050 | ( | ( | 72 | 48 | 110 | H | H | 104 | 68 | 150 | h | h |
| 9 | 9 | 011 | TAB (horizontal tab) | 41 | 29 | 051 | ) | ) | 73 | 49 | 111 | I | I | 105 | 69 | 151 | i | i |
| 10 | A | 012 | LF (NL line feed, new line) | 42 | 2A | 052 | * | * | 74 | 4A | 112 | J | J | 106 | 6A | 152 | j | j |
| 11 | B | 013 | VT (vertical tab) | 43 | 2B | 053 | + | + | 75 | 4B | 113 | K | K | 107 | 6B | 153 | k | k |
| 12 | C | 014 | FF (NP form feed, new page) | 44 | 2C | 054 | , | , | 76 | 4C | 114 | L | L | 108 | 6C | 154 | l | l |
| 13 | D | 015 | CR (carriage return) | 45 | 2D | 055 | - | - | 77 | 4D | 115 | M | M | 109 | 6D | 155 | m | m |
| 14 | E | 016 | SO (shift out) | 46 | 2E | 056 | . | . | 78 | 4E | 116 | N | N | 110 | 6E | 156 | n | n |
| 15 | F | 017 | SI (shift in) | 47 | 2F | 057 | / | / | 79 | 4F | 117 | O | O | 111 | 6F | 157 | o | o |
| 16 | 10 | 020 | DLE (data link escape) | 48 | 30 | 060 | 0 | 0 | 80 | 50 | 120 | P | P | 112 | 70 | 160 | p | p |
| 17 | 11 | 021 | DC1 (device control 1) | 49 | 31 | 061 | 1 | 1 | 81 | 51 | 121 | Q | Q | 113 | 71 | 161 | q | q |
| 18 | 12 | 022 | DC2 (device control 2) | 50 | 32 | 062 | 2 | 2 | 82 | 52 | 122 | R | R | 114 | 72 | 162 | r | r |
| 19 | 13 | 023 | DC3 (device control 3) | 51 | 33 | 063 | 3 | 3 | 83 | 53 | 123 | S | S | 115 | 73 | 163 | s | s |
| 20 | 14 | 024 | DC4 (device control 4) | 52 | 34 | 064 | 4 | 4 | 84 | 54 | 124 | T | T | 116 | 74 | 164 | t | t |
| 21 | 15 | 025 | NAK (negative acknowledge) | 53 | 35 | 065 | 5 | 5 | 85 | 55 | 125 | U | U | 117 | 75 | 165 | u | u |
| 22 | 16 | 026 | SYN (synchronous idle) | 54 | 36 | 066 | 6 | 6 | 86 | 56 | 126 | V | V | 118 | 76 | 166 | v | v |
| 23 | 17 | 027 | ETB (end of trans. block) | 55 | 37 | 067 | 7 | 7 | 87 | 57 | 127 | W | W | 119 | 77 | 167 | w | w |
| 24 | 18 | 030 | CAN (cancel) | 56 | 38 | 070 | 8 | 8 | 88 | 58 | 130 | X | X | 120 | 78 | 170 | x | x |
| 25 | 19 | 031 | EM (end of medium) | 57 | 39 | 071 | 9 | 9 | 89 | 59 | 131 | Y | Y | 121 | 79 | 171 | y | y |
| 26 | 1A | 032 | SUB (substitute) | 58 | 3A | 072 | : | : | 90 | 5A | 132 | Z | Z | 122 | 7A | 172 | z | z |
| 27 | 1B | 033 | ESC (escape) | 59 | 3B | 073 | ; | ; | 91 | 5B | 133 | [ | [ | 123 | 7B | 173 | { | { |
| 28 | 1C | 034 | FS (file separator) | 60 | 3C | 074 | < | < | 92 | 5C | 134 | \ | \ | 124 | 7C | 174 | | | \| |
| 29 | 1D | 035 | GS (group separator) | 61 | 3D | 075 | = | = | 93 | 5D | 135 | ] | ] | 125 | 7D | 175 | } | } |
| 30 | 1E | 036 | RS (record separator) | 62 | 3E | 076 | > | > | 94 | 5E | 136 | ^ | ^ | 126 | 7E | 176 | ~ | ~ |
| 31 | 1F | 037 | US (unit separator) | 63 | 3F | 077 | ? | ? | 95 | 5F | 137 | _ | _ | 127 | 7F | 177 | &#127; | DEL |

Source: www.asciitable.com

# PROGRAMMIERBEISPIELE

## QUELLCODE CRC BERECHNUNG

Im Paketmodus kann eine CRC Prüfsumme verwendet werden, um Daten bei der Übertragung zu sichern. Die Berechnung des CRCs kann nach folgendem Schema erfolgen:

```
static const unsigned int crc_tab[16] =
{
  0x0000, 0x1081, 0x2102, 0x3183,
  0x4204, 0x5285, 0x6306, 0x7387,
  0x8408, 0x9489, 0xA50A, 0xB58B,
  0xC60C, 0xD68D, 0xE70E, 0xF78F
};


unsigned short crc_update (unsigned short crc, unsigned char c)
{
    crc = (((crc >> 4) & 0x0FFF) ^ crc_tab[((crc ^ c) & 0x000F)]);
    crc = (((crc >> 4) & 0x0FFF) ^ crc_tab[((crc ^ (c>>4)) & 0x000F)]);
    return crc;
}
```

- Eine Variable (16 Bits, der CRC-Wert) wird mit 0xffff initialisiert:
  **unsigned short crc_wert = 0xffff;**

- Für jedes Byte des Pakets wird die Variable verändert, indem man die obenstehende Funktion aufruft (in einer Schleife):
  **crc_wert = crc_update (crc_wert, aktuelles_byte);**

- Nachdem das letzte Byte verarbeitet wurde, wird die Variable negiert:
  **crc_wert = ~crc_wert;**

## QUELLCODE „WRITE PACKET“

Das folgende Beispiel illustriert, wie eine Anwendung ein Paket generieren könnte, das zum Avisaro Modul übertragen wird.

```
void CommWritePacket (HANDLE hcomm, char *data, short size)
{
    unsigned short crc = 0xffff;

    CommWrite (hcomm, size & 0xff);
    CommWrite (hcomm, size >> 8);
    while (size--)
    {
        CommWrite (hcomm, *data);
        crc = crc_update (crc, *data);
        data++;
    }

    //CommWrite (hcomm, 0); //diese Zeilen verwenden, wenn kein
    //CommWrite (hcomm, 0); //CRC verwendet werden soll
    crc = ~crc;
    CommWrite (hcomm, crc & 0xff); // diese Zeilen weglassen, wenn
    CommWrite (hcomm, crc >> 8);   // kein CRC verwendet werden soll
}
```